Machine Learning 1

Lecture 9.4 - Unsupervised Learning
Gaussian Mixture Models - The Expectation Maximization Algorithm

Erik Bekkers

(Bishop 2.3.9, 9.2)

Slide credits: Rianne van den Berg

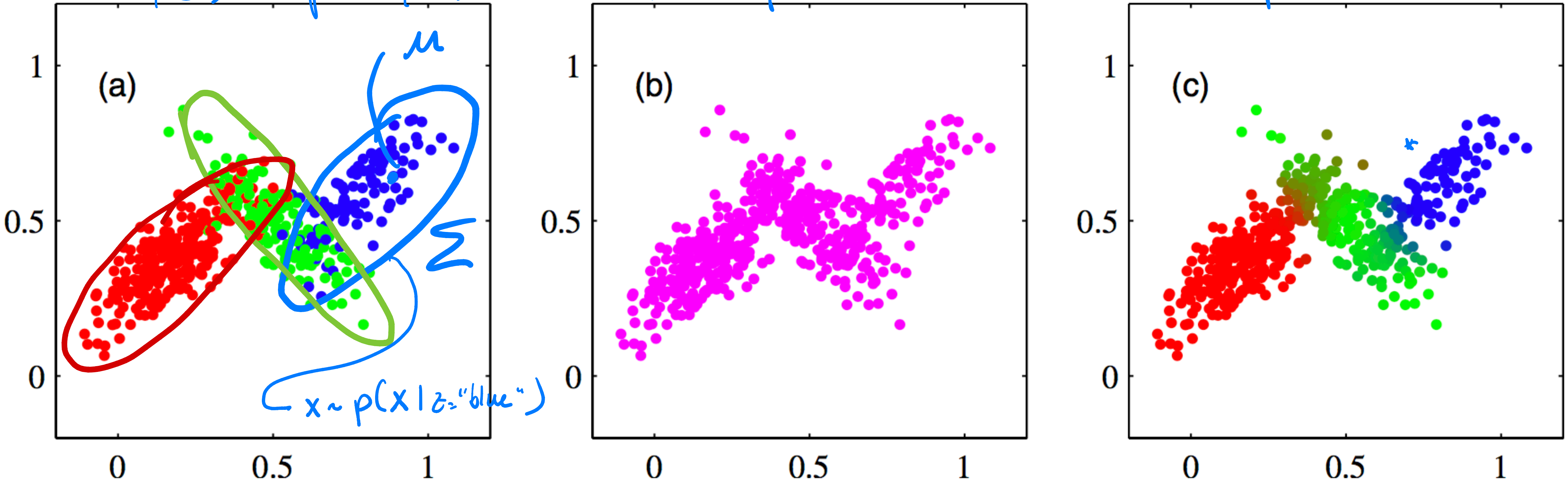
Image credit: Kirillm | Getty Images

Clustering with Gaussian Mixture Model (GMM)

- Generative model: $p(x) = \sum_z p(x, z) = \sum_z p(x|z)p(z)$
- Approximate the distribution with a mixture of Gaussians

  Bayes ⇓ $p(z|x)$
- A discrete random variable picks the cluster (the mixture z component) and points in the cluster are Gaussian distributed
- More flexible than K-means

$(x, z) \sim p(x, z)$ $\quad x \sim p(x)$ $\quad p(z|x)$

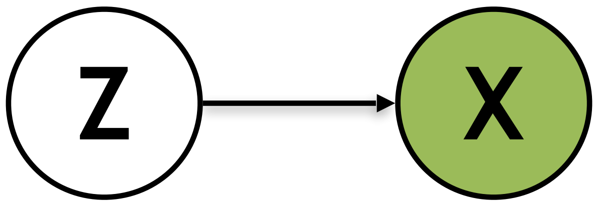


Original data | Unlabelled sample | GMM: soft-clustering

Formally

- Data: $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}, \boldsymbol{x}_n \in \mathbb{R}^D$
- Goal: partition into K clusters by maximizing the likelihood of the probabilistic model $p(x)$
- Recall the discrete latent variable model from the start

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}) = \sum_{\boldsymbol{z}} \underbrace{p(\boldsymbol{x}|\boldsymbol{z})}_{\text{Gaussians}} \underbrace{p(\boldsymbol{z})}_{\text{Generalized Bernoulli}}$$



$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$$

Modeling assumptions

- 1-hot-encoded **discrete latent variable** $z_k \in \{0,1\}$ for the K clusters, with **prior**

Constraint!

$$p(z_k = 1) = \pi_k, \pi_k \in [0,1], \sum_{k=1}^{K} \pi_k = 1$$

- The clusters are Gaussians, with different parameters

$$p(\boldsymbol{x}|z_k = 1) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- It follows that the joint is

$$p(\boldsymbol{x}, z_k = 1) = p(\boldsymbol{x}|z_k = 1)p(z_k = 1) = \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- And the marginal… the full **generative model**

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}) = \sum_{k} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

The posterior

- The conditional probability of z (the latent cluster) given a point x

$$
\begin{aligned}
p(z_k = 1|\boldsymbol{x}) &= \frac{p(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{p(\boldsymbol{x})} \\
&= \frac{p(z_k = 1)p(\boldsymbol{x}|z_k = 1)}{\sum_j p(z_j = 1)p(\boldsymbol{x}|z_j = 1)} \\
&= \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} =: \underline{\gamma(z_k)}
\end{aligned}
$$

<u>Responsibility</u> that class k takes for explaining data point x

The log-likelihood

- Given the data $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$

$$\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \overset{\text{i.i.d}}{=} \ln \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Cannot further simplify because of the sum

- How to maximize the log-likelihood?

$\longrightarrow$ EM

Expectation-Maximization algorithm (EM)

- We need to maximize the likelihood with respect to $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \forall k = 1, \ldots, K$

$$\ln p(X) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\boxed{\frac{\partial}{\partial \mu_k} \ln p(X) = 0 \quad \Rightarrow \mu_k = \ldots\ldots \gamma(z_{nk})}$$

- The problem is non-convex
- No closed-form solution! Stationary points depends on the posterior $\gamma(z_{nk})$

π_k, μ_k, Σ_k

- We can find local minima by iterative algorithm: alternate update of (**expected**) posterior $\gamma(z_{nk})$ and **maximization** for $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ (params)

Expectation-Maximization algorithm (EM)

- We need to maximize the likelihood with respect to $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \forall k = 1, \ldots, K$

$$\sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Solve with $\gamma(z_{nk})$ fixed using current estimates $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$
- $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n$ $\quad\quad$ $\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$

  $\pi_k = \frac{N_k}{N}$ $\quad\quad$ $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$

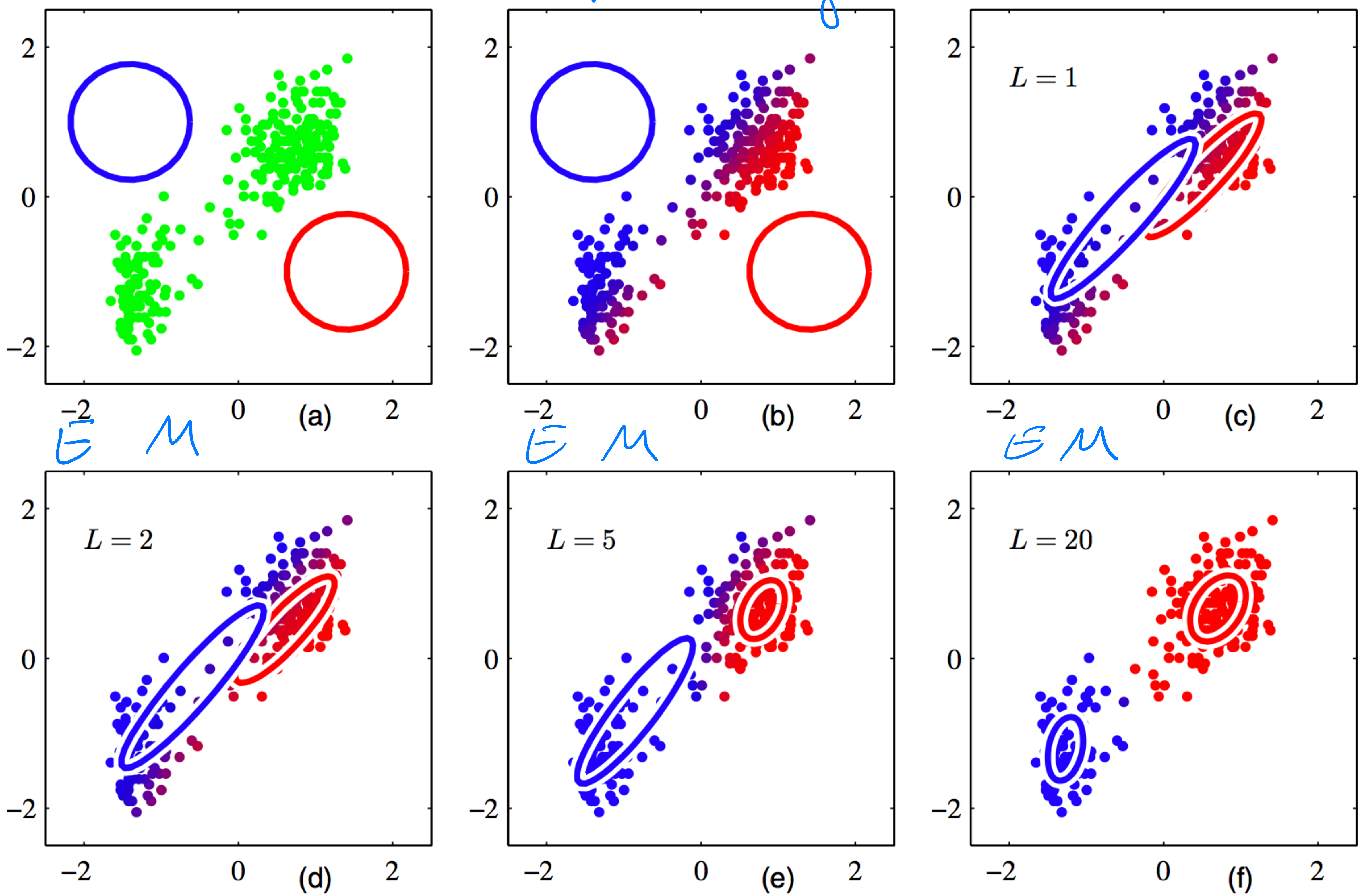
- We can find local minima by iterative algorithm: alternate update of (**expected**) posterior $\gamma(z_{nk})$ and **maximization** for $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ (params)

Example: GMM

init
π_k, μ_k, Σ_k

E: update expected posterior $\gamma(z_{nk})$

M: maximization using $\gamma(z_{nk})$



Some useful facts on multivariate Gaussians

- Multivariate Gaussian:

$$\mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)}$$

- Density derivative with respect to $\boldsymbol{\mu}_k$

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \overbrace{\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}^{\text{because of the exponent}} \underbrace{(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1}}$$

(because Σ^{-1} is symmetric)

$\Sigma^{-1} = \Sigma^{-1\,T}$

$$\frac{\partial}{\partial \mu_k}\left(\frac{1}{2}(\underline{x} - \underline{\mu}_k)^T \Sigma^{-1} (\underline{x} - \underline{\mu}_k)\right) = \frac{1}{2}(\underline{x} - \underline{\mu}_k)^T\left(\Sigma^{-1} + \Sigma^{-1\,T}\right)$$

Maximize with respect to $\boldsymbol{\mu}_k$

- Set the derivative wrt $\boldsymbol{\mu}_k$ of the log-likelihood to 0

$$\frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^{N} \log p(\boldsymbol{x}_n \mid \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\})$$

$$= \sum_{n=1}^{N} \frac{1}{p(\boldsymbol{x}_n \mid \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\})} \frac{\partial}{\partial \boldsymbol{\mu}_k} p(\boldsymbol{x}_n \mid \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\})$$

$$= \sum_{n=1}^{N} \boxed{\frac{\pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}$$

μ_k = the weighted average over the points x_n for which cluster k takes responsibility

$$= \sum_{n=1}^{N} \boxed{\gamma(z_{nk})} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} = 0 \quad \Rightarrow \quad \boxed{\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})\, \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})}}$$

Maximize with respect to π_k

Constraint $\underbrace{\sum_k \pi_k}_{g(x)} = \underbrace{1}_{c}$

Lagrange multipliers

- Set the derivative w.r.t. π_k of the log-likelihood to 0

$$\frac{\partial}{\partial \pi_k}\left(\overbrace{\sum_{n=1}^{N} \log p(\boldsymbol{x}_n \,|\, \{\pi_k\}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\})}^{f(x)} + \overbrace{\lambda \left(\sum_{j=1}^{K} \pi_j - 1 \right)}^{\lambda(g(x) - c)} \right)$$

$$= \sum_{n=1}^{N} \overbrace{\boxed{\frac{\mathcal{N}(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n \,|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}}^{\frac{1}{\pi_k}\gamma(z_{nk})} + \lambda = 0 \quad \Rightarrow \quad \pi_k = -\frac{1}{\lambda} \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\sum_{j=1}^{K} \gamma(z_{nj}) = \sum_{j=1}^{K} p(z_{nj}=1 \,|\, x_n) = 1$$

$$\frac{\partial}{\partial \lambda} L(\{\pi_k\}, \lambda) = \sum_{j=1}^{K} \pi_j - 1 = -\frac{1}{\lambda} \sum_{j=1}^{K} \sum_{n=1}^{N} \gamma(z_{nj}) - 1 = 0$$

$$\lambda = -N \quad \Rightarrow \quad \boxed{\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})}$$

fraction of points for which cluster k takes responsibility

Equations for the M-step

- Define, the "effective number of points in cluster k" by

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

- Solutions for $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ (dependent on the posterior)

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \qquad \pi_k = \frac{N_k}{N}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

The EM algorithm for GMM

- Initialize with a random $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$
- Repeat until convergence:
  - Update the posterior – **Expectation-step**

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

  - Update the parameters – **Maximization-step**

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \qquad \pi_k = \frac{N_k}{N}$$

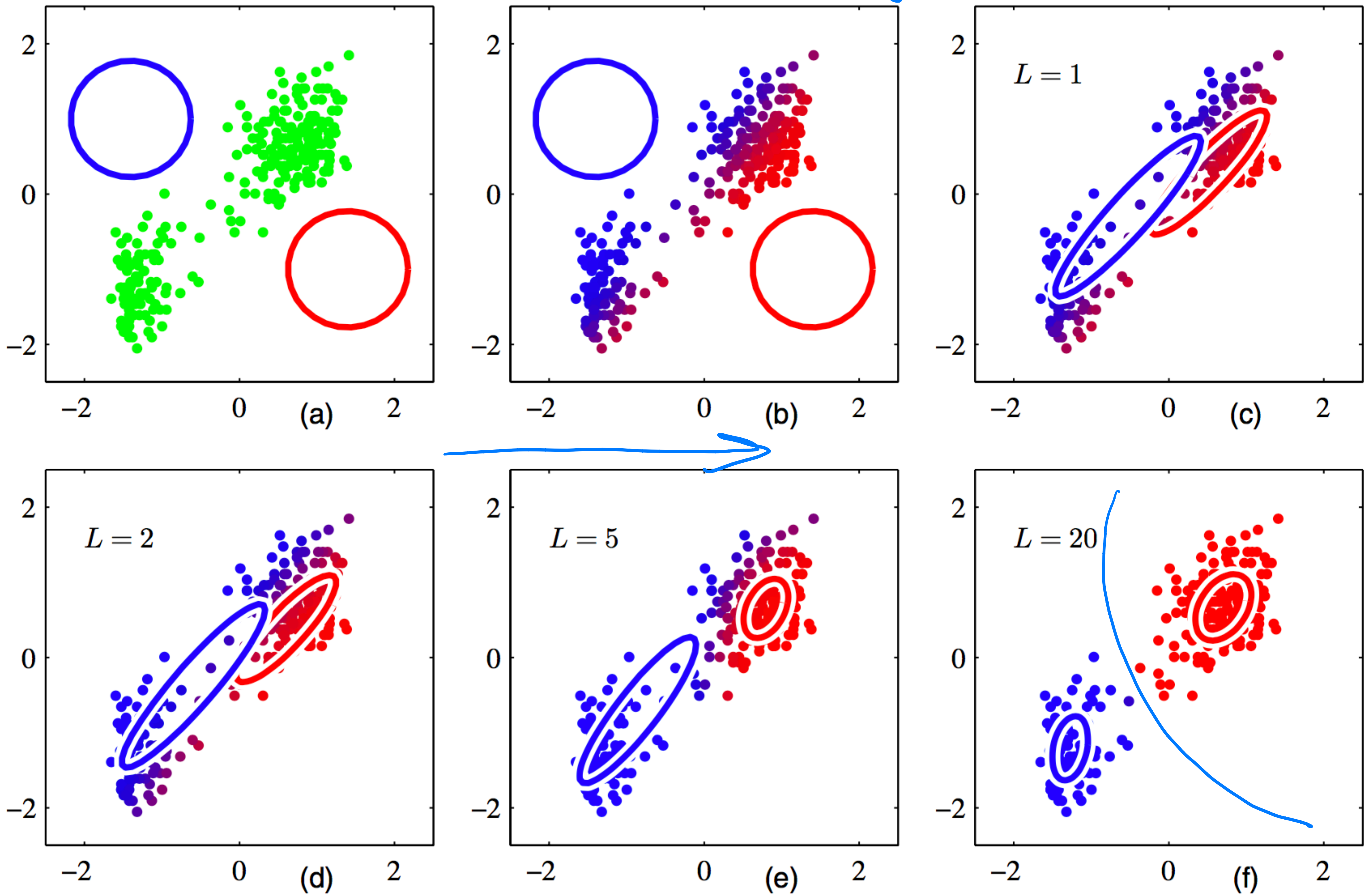
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\top$$

Example: GMM

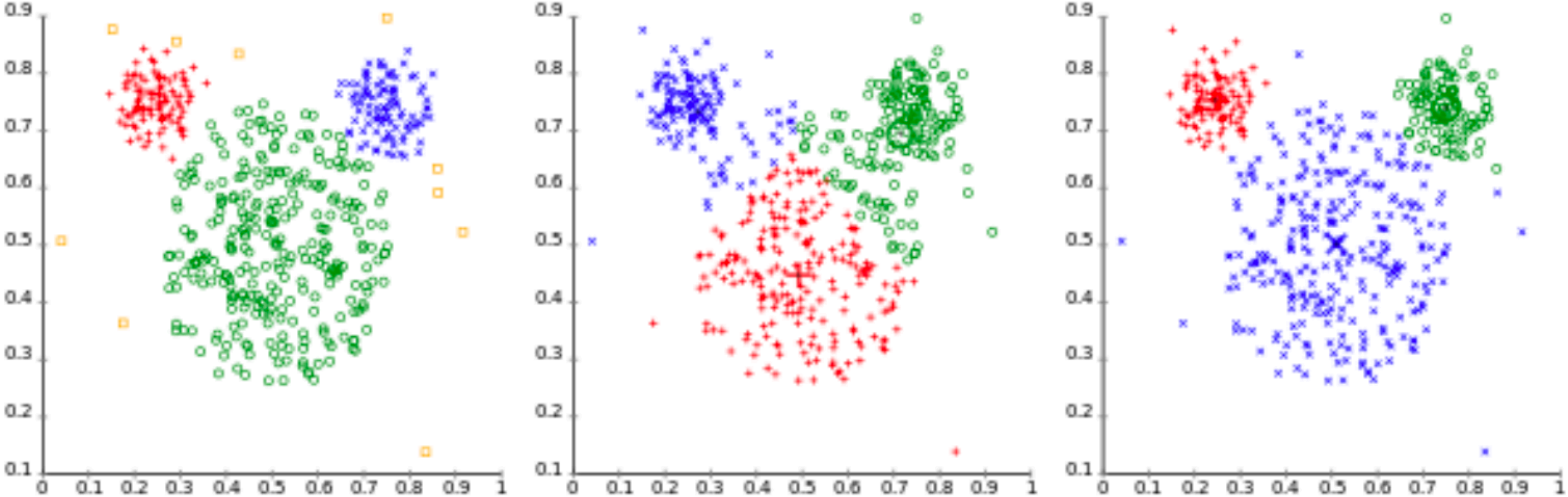
init : π_k, μ_k, Σ_k

E : soft assignments based on $\gamma(z_{nk})$

M



The mouse data again



original clusters K-means GMM

- K-means ignores different covariance of the clusters. GMM can model those differences.

How do we assign points to clusters?

Soft-clusters

- The posterior tells us the probability of belonging to every possible cluster k

$$\gamma(z_k) = p(z_k = 1 \mid \underline{x})$$

And if you need hard-clusters:

- The most likely cluster is given by

$$k = \underset{j=1,\ldots,K}{\operatorname{argmax}} \gamma(z_j)$$

Comments

- GMM gives soft-assignments in contrast with K-means
- GMM is more flexible because we can model a different covariance per cluster
- GMM is slower than K-means. We can use K-means to initialize the cluster means
- Same local convergence issues as for K-means
- GMM is the similar to Quadratic Discriminant Analysis, but the target is unknown and we use the EM algorithm for learning